UNIVERSITY OF OSLO
Department of Informatics

Personal device integration, content access and simple pairing procedures

Masteroppgave

Øystein Sandnes

22. Januar 2008

Abstract

Personal device integration is the process of interconnecting devices, such as mobile phones and home entertainment systems. In this thesis we examine the underlaying technology, the need for development and current progress in this field. We created a test suite in order to gain a better understanding of the difficulties surrounding this technology, and to see what functionality would be possible using the current technology. In order to get a broader view of this project, testing was done using different kinds of underlaying software and operating systems. These include Linux, Microsoft Windows XP and Vista, and the mobile operating systems S60 and Windows Mobile.

1 Acknowledgements

I would like to thank Prof. Josef Noll who has been a great inspiration and support. His advice has made this paper possible. I also want to thank György Kálmán, who has given me advice and support, but also as a friend. I also like to thank the following people for beeing there for me in my time of personal need: Andreas Svendsen, Bendik W. Andreassen, Niels Petter Rasch-Olsen, Erik Lindvig, Lars Høiby, Audun Andersen, Karianne Olsen, Athea Olsen, Mona Irene Wensaas, Oddmund Broen, Joachim Øhlschlägel, Jostein Malde, Jan Sigurd Knap, Jan Tore Jensen, Ragnar Jensen, Jørn Hammerud, Jørgen Hovland, Hans-Christian Lindvig and my familiy, Ludvik, Jorunn and Ida Sandnes.

Thank you all.

Contents

1 Acknowledgements **2**

2 Motivation **6**

3 Introduction **7**

4 Scenario - Case Studies **8**
4.1 Multi Purpose Car Phone 8
4.2 Set-Top-Box Integration 9
4.3 Secure authentication 10

5 Current Technological Status **11**
5.1 Radio Frequency Identification - RFID 11
5.2 NFC 12
5.3 Smartcard 13
5.4 WIFI - ieee 802.11b 14
5.5 Wimax 15
5.6 Bluetooth 15
5.6.1 Pairing in Bluetooth 16
5.6.2 Bluetooth stack architecture 16
5.6.3 Bluetooth protocols 17
5.7 Simple Pairing and cryptosystems 19
5.7.1 Public-key cryptosystems 19
5.7.2 Diffie-Hellman Key Exchange 20
5.7.3 Eliptic Curve 22
5.7.4 Simple Pairing Objectives - Security 22
5.8 Wireless Security - Bluetooth and WIFI 23
5.8.1 WEP, WPA and WPA2 23
5.8.2 Bluetooth Security 24

6 Easy Pairing in Practice **26**
6.1 Easy Connection Set-up 26
6.2 Service Examples 30
6.2.1 Bookmarks and Shortcut Link Distribution 31
6.2.2 Client Software Upload 31
6.2.3 Authentication and Access Control 32
6.3 Media Center 32
6.4 A Linux Based Appliance 33
6.4.1 Hardware Extensions 34
6.4.2 Network Connectivity 35

6.4.3 Local Services 35
6.4.4 Streaming 36
6.4.5 Easy Pairing on Dreambox 37
6.4.6 Drawbacks 37
6.5 Bluetooth PAN and Personal Content Access 38
6.6 Personal Content Access through PAN in relation to Windows Media Center 39

7 Implementation Issues **41**
7.1 Pin Request Dialog using Java, Bluetooth and Windows . . . 41
7.2 Security Risks in Current Implementation 42
7.3 NFC Tags and Devices 42

8 Conclusion **43**

9 Reference **44**

List of Figures

1 Your phone as a universal remote and communication device . 9
2 With secure authentication we can explore several different fields of secure transactions 11
3 802.11 standards overview 14
4 Bluetooth stack architecture[6] 16
5 Bluetooth protocol stack 17
6 Simple pairing association model[12] 24
7 Media server with contactless reader 27
8 By attaching these chips to a Bluetooth enabled device we can enable easy pairing. 28
9 Nokia 6131 and RFID chip 28
10 Triggering easy pairing by placing phone next to contactless reader 29
11 Method to handle discovered devices 29
12 Requesting pin input from user to complete pairing 30
13 Webguide user interface screenshot taken from the WebGuide Documentation 40
14 Streaming video from media center to a PPC 41

2 Motivation

When I first started this thesis my main goal was to ease communciation between devices, explore the possibilities surrounding this subject, and later on take personal content access into account. This brought me to the subject of easy pairing and short range contactless communication.

My first step was to analyse technologies and evaluate potential for easy pairing. This phase also included an investigation of which players where currently active in the marked, and establishing contact to exchange information.

Second step was to give access to personal content. After some months of work we recognised that Microsoft has recently purchased a solution from a application development company which created a plugin for Windows Media Center[1]. As this happened after the already performed work of the initial pairing we understood that our research was really in the focus of digital convergence.

Due to these ongoing commerial development we slightly changed focus of the thesis and concentrated on the simple pairing procedure as a gateway to service and content access.

My tasks were:

- Technology review
- Scenario descriptions
- Establish a knowledge of the software development kit (SDK) for Nokia 3320 and 6131
- Easy pairing
- Personalized access

This work was done in close co-operation with the international ITEA Wellcom project, where results will be used for distribution of personalised content to mobile phones. Amongst the 24 Wellcom partners are Telenor, SES Astra, TF1 and other important media and telecom companies.

[1] WebGuide for Windows Media center - http://www.asciiexpress.com/webguide/

3 Introduction

With the ever increasing popularity of home-multimedia appliances, people have gotten used to accessing multimedia content where ever and when ever they want. To be able to do this, we need some way of making these devices all work together and share information. With the combination of WIFI, Bluetooth and Ethernet we can provide a solution for most of these needs, but this normally demands some experience in computer networking. One solution would be to simplify the process of inter connecting these devices, but this would have to be done without compromising security. One potential solution is to use Near Field Communication (NFC section 5.2) to setup the initial connection. This connection can be used to set up a permanent connection between the devices allowing them to share information or services. For added security, the information exchanged during the initial setup can be used to authenticate a user to an appliance (i.e. Digital subscription or Digital tv-tuner).

If we use a Bluetooth and NFC enabled digital tv-tuner and use a Bluetooth and NFC enabled mobile phone, we can use a NFC connection to establish Bluetooth pairing (described in section 5.6) and by using the information programmed onto the NFC chip we can initiate a Bluetooth pairing enabling the user to take control over the appliance hence, use the mobile-phone as a remote. The pairing of these devices will also make it possible to supply other services such as the transfer of an Electronic Program Guide (EPG) directly to the phone.

This thesis is composed in the following manner: We will first introduce some of the concepts and terminology used to examine the example described above. Then in chapter 4 we will look more deeply into the areas where this technology could be applied. Finally we will show some background information conserning the different players in the market, and their current solutions in chapter 5, see a practicle example in chapter 6 before we take a look at future work, and conclusions in chapter 8.

4 Scenario - Case Studies

The following section will examine some scenario cases where this technology would be applicable. We will also discuss some of the current technological solutions to support these scenarios.

4.1 Multi Purpose Car Phone

The mobile phone should be able to interact with your car. When you get into the car and your phone gets within reach of the sensory equipment of the car, a wireless pairing procedure should be initiated between your mobile phone and the driving computer system. After the initial setup is completed, the car will adapt to your prefered settings, including your favorite radio-channels, seat and mirror adjustments, and then run a local update of GPS roadmap data using the phone as an Internet connection. In addition you should be able to use the handsfree system embedded in the car to read messages, access the phone book and place a call.

There are currently several players in this market, including major car companies such as BMW and Audi who both have included car phones and GPS tracking and navigation system in their more expensive models. Since most built-in car phones lack functionality like Bluetooth, Personal Information Management and multimedia services most users will have a second phone from another company like Nokia or Sony-Ericsson. To eliminate the need for two seperate subscriptions or seperate phone numbers, the service providers have opened up for a solution known as "twin-sim"[20]. This basically means that a subscriber can be reached on two seperate phones using the same number, but with two different sim-cards, enabling the subscriber to use one in his personal phone and one in the integrated car-phone system. There are two major problems with this solution. The first one is keeping internal userdata like addressbook and schedule synchronised between the two devices. The second problem comes from the fact that even though both sim-cards respond to the same number during calls, they still need to use a different service access number. Having two Service Access Numbers gives the user two different identities in the cellular network.

Some manufacturers are starting to include Bluetooth support which can connect to a Bluetooth enabled phone, providing handsfree access and syncronizing phone book etc, but you still have to go through a pairing procedure. Easy pairing would make this much simpler. University Graduate Center at Kjeller (UniK) has an ongoing discussion with BMW and NEC on how simple pairing might be implemented in the car communication system.

Figure 1: Your phone as a universal remote and communication device

4.2 Set-Top-Box Integration

Imagine Tom and Lisa. They both have a digital tv subscription with authentication and decoding provided by a smartcard. Tom prefers the movie channel, and Lisa got the sports channel. Tom and Lisa wants to share a quiet evening together and they have decided to see a movie and a football match. Since Lisa got the sports channel they can watch the game as planned, but they would have to replace the smartcard in Lisa's decoder with Toms' movie channel smartcard to see the movie. The problem is that Tom forgot his decoder card at home. What we actually want is illustrated in figure 1. We want the phone as a universal remote and communication device.

This is no longer a problem because Tom can access his content remotely. To enable this functionality we introduce the NFC and Bluetooth capable mobile phone and tv-tuner. Tom's subscription and authentication information is contained as a certificate on his mobile phone, and when the game is finished he holds his phone up against the casing of Lisa's NFC and bluetooth-ready digital subscription tuner.

The phone and tuner immediately establish a connection and the phone transfers a temporarily certificate to the tuner. This unlocks the movie channel for 24 hours using Tom's subscription. The tuner then transfers the

scheduled program guide to Tom's phone and continue to set up a secure Bluetooth pairing. Tom accepts the Bluetooth pairing and by doing so also accepts the transfer of the remote control service to his mobile phone. He can now control the tuner device with his mobile phone and using the EPG which he received earlier they decide on a movie and change to the appropriate channel. Neither of them have seen this movie before so still using the EPG on his mobile phone, he follows a link to the Internet Movie Database to read more about it. Every underlaying technology is present, but we still lack the initial setup, which could be completed by using simple pairing. One part of this scenario requires authentication, which is discussed in the next scenario.

4.3 Secure authentication

Imagine not having to carry a wallet, credit card or keys to your car. If we can utilize the mobile phone to ensure secure authentication we can make this happen. If we make the SIM card and operating status of the mobile phone into a part of the authentication, we can add an extended security function as opposed to normal keys, because both phone and sim-card can be remotely disabled. But for this technology to gain wide acceptence and trust within the public, we need a way of making it easy to use while keeping the security aspects.

Several players in the market have started research and development of their own secure contactless solutions (Mastercard, VISA and subway ticket companies as illustrated in figure 2). This will however not eliminiate the need for bringing a credit card or other form of identification, thus leaving us at the same situation than we are today.

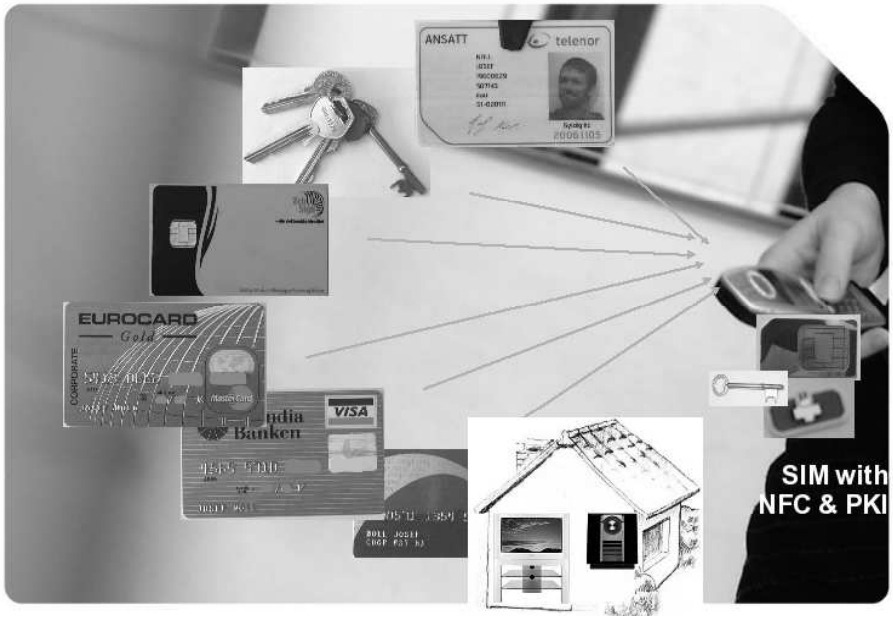

Figure 2: With secure authentication we can explore several different fields of secure transactions

5 Current Technological Status

This section will give an understanding of the underlaying technology that is currently in use, or beeing developed today, and has been a subject during the research phase of this thesis.

5.1 Radio Frequency Identification - RFID

RFID is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders. An RFID tag is an object that can be attached to or incorporated into a product, animal, or person for the purpose of identification using radio waves. Chip-based RFID tags contain silicon chips and antennas. Passive RFID tags have no internal power supply. The electrical current induced in the antenna by the incoming radio frequency signal provides power for the CMOS integrated circuit in the tag to power up and transmit a response. Using an onboard power supply, active tags transmits at higher power levels than passive tags, allowing them to be more effective in "RF challenged" environments like water, metal (shipping containers, vehicles), or when used inside humans/cattle and at longer distances (typical at data rates of 106 to 848 kbit/s)[14].

5.2 NFC

NFC is a RFID variant for wireless communication which at the time of writing is under development. It is an early stage offspring of the smartcard technology found in Sony FeliCa and Philips MIFARE, and operates at 13.56MHz. The purpose of this technology is to enable short-term (ad-hoc) communication and/or authentication between different types of personal devices, just by holding them close together [9].

The magnetic field can be used for contactless short range communications. Consider NFC a transformer with a very low coefficient of coupling because of a large distance between primary (transmitter) and secondary (receiver) windings (antennas). According to Louis E.Frenzel of Electronic Design[9] the signal strength drops off at a rate of about 1/d6, where d is the distance or range, truly making it a short range technology. Philips and Sony invented NFC[2] and Ecma International adopted it as a standard first (NFCIP-1 or ECMA-340) and submitted it to the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC), which standardized it as ISO/IEC 18092 [22]. Later on the European Telecommunications Standards Institute (ETSI) also has accepted NFC as a standard and semiconductor companies have since then begun making compatible and interoperable chips.

This standard is similar to and compatible with the same NFC technology used in smartcards, whose internal chip lets consumers pay by passing them over a Point of Sale (PoS) terminal reader. In some modes, NFC also resembles RFID described in section 5.1. The well established smartcard standard (ISO 14443), is implemented in Philips' MIFARE and Sony's FeliCa products. The standard specifies an operating frequency of 13.56 MHz, the international no-license band. The data transfer rate is 106, 212, or 424 kbits/s. The speed depends upon the range, which has a aproximate maximum of 20 cm. In most cases, the actual range will be only a few inches or no more than 10 cm [22]. Also, the standard specifies several operational modes. In the active mode, both parties have powered transceivers. Either party may initiate a half-duplex transmission with a "listen before transmit" protocol. This feature prevents collisions when more than one NFC-enabled device tries to access a reader. One of the devices is the initiator, and the other device becomes the target [22].

In the passive mode, the target is a passive device like an RFID tag. The tag gets its operational power from the field transmitted by the initiator. When it receives such a call it can respond by returning a modulated version of the senders request, or create an event triggering an internal (software or

[2]For more information see NFC Forum - http://www.nfc-forum.org/home

hardware) function in the NFC-device i.e. opening a communication channel or initiate an authentication handshake, data-transfer or a more complex form of networking like Bluetooth or WIFI.

As with any new wireless technology, security is an issue. But the very short range of NFC devices makes eavesdropping less of an issue. At that distance, all you have to do is show intent and you're safe for the most part. To add more security, NFC can be used in combination with encryption technology known from smartcard (see section 5.3). The limited range of the devices have eliminated the need for power supplies, making NFC capable devices small and compact. This has also made it possible to fit a NFC chip on to almost any kind of device which again makes it very useful when it comes to making devices work together.

NFC or RFID can in general be used for content distribution. An example could be adding RFID chips to a bus-stop containing information or linking to information about the current time table or delay time. The information can be in the form of a URL pointing to a preconfigured website, or if the data is sufficiently small, all of the information can be contained within the RFID chip. This will make it possible to obtain this information using your NFC capable phone.

5.3 Smartcard

A smartcard, chip card or integrated circuit(s) card, is defined as any pocket-sized card with embedded integrated circuits [21]. This is normally a credit-card sized card with various tamper-resistant properties (e.g. a secure crypto-processor or an encrypted file system.) and is therefore capable of providing secure authentication or other security services (like payment procedures).

There are two main types of smartcards, contact based smartcard and contactless smartcard. In this case we concentrated on the contactless technology. In contactless smartcard technology, communication with the card reader is done by using RFID (see section 5.1) induction technology. These cards require only close proximity to an antenna to complete transaction. They are often used when transactions must be processed quickly or hands-free, such as on mass transit systems, where smartcards can be used without even removing them from your wallet.

Smartcard technology is currently used for secure transactions (payment), access control on digital tv-tuners and signing of official documents using your computer and an Internet connection[3].

[3]More information on the Mastercard homepage - http://www.mastercard.com

Protocol	Release Date	Op. Frequency	Throughput (Typ)	Data Rate (Max)	Modulation Technique	Range (Radius Indoor) Depends, # and type of walls	Range (Radius Outdoor) Loss includes one wall
Legacy	1997	2.4 GHz	0.9 Mbit/s	2 Mbit/s		~20 Meters	~100 Meters
802.11a	1999	5 GHz	23 Mbit/s	54 Mbit/s	OFDM	~35 Meters	~120 Meters
802.11b	1999	2.4 GHz	4.3 Mbit/s	11 Mbit/s	DSSS	~38 Meters	~140 Meters
802.11g	2003	2.4 GHz	19 Mbit/s	54 Mbit/s	OFDM	~38 Meters	~140 Meters
802.11n	June 2009 (est.)	2.4 GHz 5 GHz	74 Mbit/s	248 Mbit/s	MIMO	~70 Meters	~250 Meters
802.11y	June 2008 (est.)	3.7 GHz	23 Mbit/s	54 Mbit/s		~50 Meters	~5000 Meters

Figure 3: 802.11 standards overview

5.4 WIFI - ieee 802.11b

WIFI is a standard for wireless communication which also goes by the more technical name 802.11. The name WIFI is a brand name invented in the process of marketing this standard, and is now used to describe appliances as compatible with this form of wireless communication. WIFI allows LANs to be deployed without cabling for client devices, typically reducing the costs of network deployment and expansion. Spaces where cables cannot be run, such as outdoor areas and historical buildings, can host wireless LANs. As of 2007 wireless network adapters are built into most modern laptops. The price of chipsets for WIFI continues to drop, making it an economical networking option included in ever more devices. WIFI has become widespread in corporate infrastructures, which also helps with the deployment of RFID technology (section 5.1)[4]. IEEE 802.11 specifies a 2.4 GHz operating frequency with data rates of 1 and 2 Mbps using either Direct Sequence Spread Spectrum (DSSS) or Frequency Hopping Spread Spectrum (FHSS)[1]. There are several underlaying variations of 802.11 (802.11a/b/g/n). The differences of these variations are shown in figure 3.

[4]Additional information can be found at Making business sense of real time location systems (RTLS) - http://www.rfidradio.com

As an example, IEEE 802.11b provides 5.5 and 11 Mbps raw data rates in addition to the 1 and 2 Mbps rates provided by 802.11. To provide the higher rates, 8 chip Complementary Code Keying (CCK) is employed as the modulation scheme. CCK is an M-ary Orthogonal Keying modulation [5][1] where one of M unique (nearly orthogonal) signal codewords is chosen for transmission. The spread function for CCK is chosen from a set of M nearly orthogonal vectors by the data word. CCK uses one vector from a set of 64 complex (QPSK) vectors and 6 bits to encode the code sent, this increase the speed of the 802.11 by 6 [1].The chipping rate is 11 MHz, which is the same as the DSSS system as described in the 802.11[1], thus providing the same occupied channel bandwidth.

5.5 Wimax

Worldwide Interoperability for Microwave Access (WiMAX), is a telecommunications technology aimed at providing wireless data over long distances. It is built on the 802.16 standard[2], just as WIFI was built on the 802.11 standard. WiMAX is the brandname invented by the WiMax forum, in an attempt to provide a brandname certification of compatibility to the hardware developers, the same way WIFI was for the 802.11 variations.

5.6 Bluetooth

Bluetooth is an industrial specification for wireless personal area networks (PANs), also known as IEEE 802.15.1. Bluetooth provides a way to connect and exchange information between devices such as personal digital assistants (PDAs), mobile phones, laptops, PCs, printers, digital cameras and video game consoles via a secure, globally unlicensed short-range radio frequency [6]. The Bluetooth specification is the result of cooperation by many companies under the Bluetooth SIG umbrella. It defines the system from the radio to the application level, including the software stack. The Bluetooth protocol stack is defined as a series of layers somewhat analogous to the Open Systems Interconnect (OSI) standard reference for communication protocol stacks. In this thesis we have considered Bluetooth version 2.0 and earlier, but at time of writing the first Bluetooth 2.1 products are in development an expected to be released in February 2008[11]. First we will establish a knowledge of the pairing process and then we will give a brief overview of the Bluetooth stack architecture, and take a look on some of the more commonly used Bluetooth protocols.

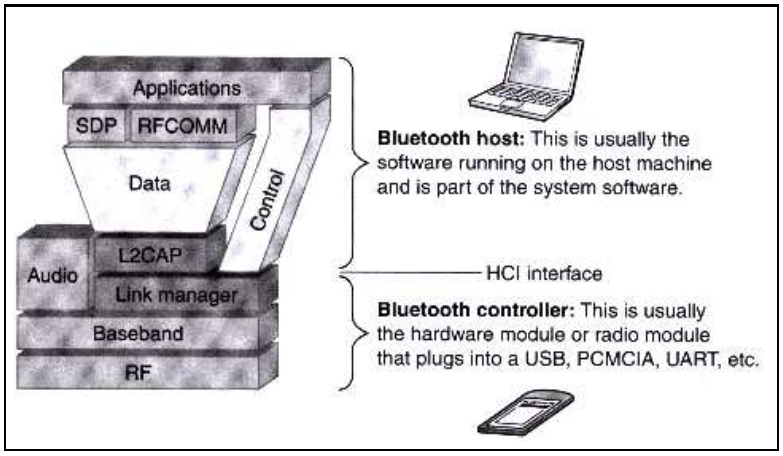

Figure 4: Bluetooth stack architecture[6]

5.6.1 Pairing in Bluetooth

When two devices establish a connection it is called a pairing, and as the name suggests, this is a private connection between two devices for the purpose of sharing some kind of information. The pairing of two devices is a trusted relationship and is established by the existence of a shared secret (key or passphrase). The two devices verify this passkey and if successful exchange information such as device name, device class (printer, phone, computer) and a list of services offered by this device. Once the devices has been paired, they will have access to all of the services.

5.6.2 Bluetooth stack architecture

When discussing the Bluetooth stack, we can divide it into two major components; the Bluetooth host, and the Blueooth controller (or radio module). This is illustrated in figure 4. In between we find the Host Controller Interface (HCI) which provides a standarized interface between the two, but in some devices (like Bluetooth headset) these two are integrated thus eliminiating the need for the HCI.[16]

The Bluetooth host or the upper layer stack, is usually implemented in software, and often integrated with the host operating system. The controller component and Bluetooth radio module is usually a hardware module like a usb dongle or integrated mini-PCI card. This module interfaces with the Bluetooth upper-layer stack through a standard input/output (I/O) mechanisms, such as Universal Serial Bus (USB) or mini peripheral component interconnect (mini-PCI).

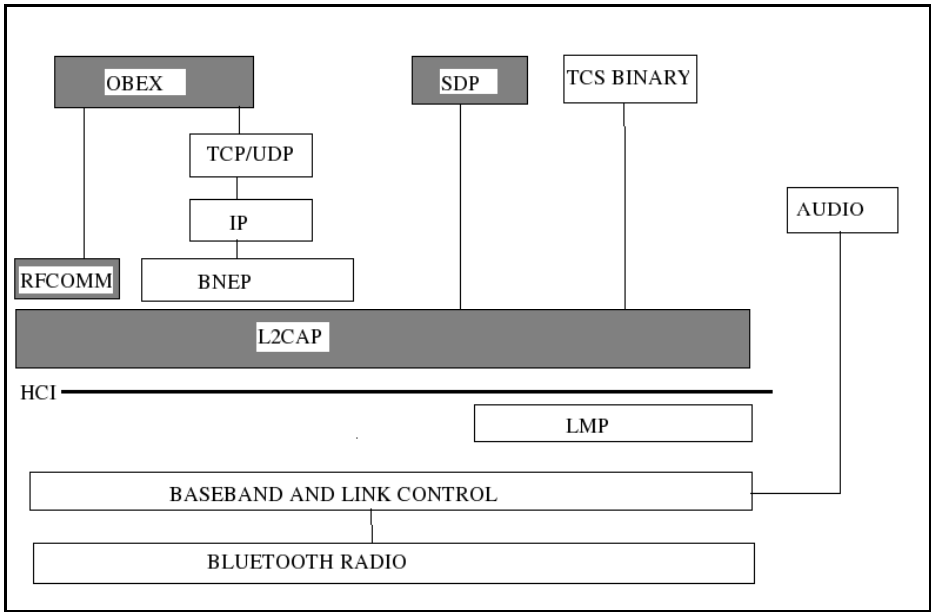

Figure 5: Bluetooth protocol stack

5.6.3 Bluetooth protocols

The protocol stack is composed of protocols specific to Bluetooth wireless technology, such as the Service Discovery Protocol (SDP) and the Object Exchange protocol (OBEX™). This is illustrated in figure 5. The shaded boxes represent the protocols addressed by Java API for wireless technology (JABWT).

The Bluetooth radio layer is the lowest defined layer in the Bluetooth specification. It operates in the 2.4GHz The industrial, scientific and medical radio band (ISM).

Baseband and link control This layer controls the physical RF link bewteen connecting Bluetooth units. The baseband handles channel processing and timing, and the link control handles the channel access control. There are two different kinds of physical links [6]: Synchronous connection oriented (SCO) and Asynchronous ConnectionLess (ACL). An ACL link carries data packets, wheras an SCO link supports real-time audio traffic.

The audio box, is showed here because it is uniquely treated in Bluetooth communication by typically routing it directly to and from the baseband layer over an SCO link. In VoIP applications or other

applications where a data channel is used to transmit audio, this would be done over an ACL link[6].

The link setup and link configuration is done by using Link Manager Protocol (LMP). The LMP manages the security aspects such as authentication and encryption, and checking link and encryption keys. This is examined more closely in section 5.8 on page 23.

The Host Controller Interface (HCI) showned in figure 5 on the preceding page as a divider between the upper and lower layers, is used to access the Bluetooth baseband capabilities, hardware status and control registers. To establish this, it provides an interface to the radio, baseband and link manager.

Logical Link Control and Adaption Protocol (L2CAP) shields the upper layer protocols from the details of the lower layer protocols. It also multiplexes between the various logical connections made by the upper layers (see section 5.6).

The Service discovery protocol (SDP) defines how a Bluetooth client's application shell acts to discover available Bluetooth servers' services and their characteristics. The protocol defines how the client can search for a service based on specific attributes without knowing anything about the available services. The SDP also provides functionality for detecting when a service is no longer available [23].

SDP allows this in various means. Searching means looking for specific service, while browsing means looking to see what services are actually being offered. The Service Search transaction allows a client to retrieve the service record handles for particular service records based on the values of attributes contained within those service records. The capability search for service records based on the values of arbitrary attributes is not provided. Rather, the capability is provided to search only for attributes whose values are Universally Unique Identifiers (UUIDs). In SDP, the mechanism for browsing for services is based on an attribute shared by all service classes. This attribute is called the BrowseGroupList attribute and the value of this attribute contains a list of UUIDs. Each UUID represents a browse group with which a service may be associated for the purpose of browsing. When a client desires to browse an SDP server's services, it creates a service search pattern containing the UUID that represents the root browse group. SDP is built on top of L2CAP.

The RFCOMM protocol provides emulation of serial ports over L2CAP, transport capabilities for upper-level services that use a serial interface as a transport mechanism. OBEX is one protocol that is built on top of RFCOMM.

Bluetooth encapsulation protocol (BNEP). Bluetooth enabled devices can establish networks and exchange information, but to do so they need a common packet format, to encapsulate layer 3 network protocols. The BNEP [23] encapsulates packets from other network-protocols and the encapsulated packets are transmitted over L2CAP. This is an optional protocol based on the 1.1 version of the Bluetooth specification. Support for the Internet Protocol over Bluetooth is done using BNEP [6].

Bluetooth is used mainly for short range communication to devices, i.e. connecting devices to mobile phones.

5.7 Simple Pairing and cryptosystems

Since almost every Bluetooth transmission starts with pairing two devices, a simple pairing technique using NFC(section 5.2) is currently beeing researched for development. The next generation of Bluetooth pairing (called Simple Pairing) will also contain a harder encryption protocol using Diffie-Hellman key exchange, and eliptic curve public key cryptography to avoid passive eavesdropping and man-in-the-middle attacks. We will look into some aspects of key exchange protocols and the simple pairing procedures as they are described in "Bluetooth special interest groups" Simple Pairing whitepaper [12].

We will now give a short introduction to public key cryptosystems and examin the diffie-hellman key exchange.

5.7.1 Public-key cryptosystems

Public key cryptography, also known as asymmetric cryptography, is a form of cryptography in which a user has a pair of cryptographic keys - a public key and a private key. The private key is kept secret, while the public key may be widely distributed. The keys are related mathematically, but the private key cannot be practically derived from the public key. A message encrypted with the public key can be decrypted only with the corresponding private key.

A public key encryption scheme has six major components:

Plaintext: This is the readable message or data that is fed into the algorithm as input.

Encryption algorithm: The encryption algorithm performs variuos mathematical transformations on the plaintext.

Public and Private key: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the encryption algorithm depend on the public or private key that is provided as input.

Cipher text: This is the scrambled message produced as output. It depends on the plaintext and the key used as well as the algorithm. For any given message, it should be such that two different keys will produce two different ciphertexts[18].

Decryption algorithm: This algorithm accepts the ciphertext and the matching key to produce the original plaintext message.

The essential steps are the following:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages. One public and one private key.

2. Each user places his or hers public key in a public register or other accessible file. The private key is kept secret.

3. If User A wants to send a confidential message to User B, User A must encrypt the message using user B's public key.

4. When User B recieves the message he or she can decrypt it using his or hers private key. No other recipient can decrypt the message because it can only be decrypted with the companion key of the given key pair.

5.7.2 Diffie-Hellman Key Exchange

The first published public key algorithm appeared in the seminal paper by Diffie and Hellman that defined public key cryptography[7], and is generally referred to as a Diffie-Hellman key exchange. The purpose of the algorithm is to enable two users to exchange a key securely that can be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of

the keys, but is designed to let users exchange keys securly without sharing a common secret.

The algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Briefly, we can define the discrete logarithm in the following way:

First, we define a primitive root of a prime number P as one whose powers generate all the integers from 1 to p - 1. That is, if a is a primitive root of the prime number p, then the numbers

$$a \bmod p, a^2 \bmod p, ..., a^{p-1} \bmod p$$

are distinct and consist of the integers from 1 through p -1 in some permutation. For any integer b and a primitive root a of prime number p, we can find a unique exponent i such that

$$b = a^i \bmod p \text{ where } 0 <= i <= (p-1)$$

The exponent i is referred to as the discrete logarithm, or index, of b for the base a mod p. This value is denoted as

$$ind_{a,p}(b)$$

[18][7]. With this background we can define the Diffie-Hellman key exchange. There are two publicly known numbers: a prime number q and an integer s that is a primitive root of q. Suppose the users A and B wish to exchange a key. User A selects a random integer

$$X_A < q$$

and computes

$$Y_A = s^{X_A} \bmod q$$

Similarly, user B independently selects a random integer

$$X_B < q$$

and computes

$$Y_B = s^{X_B} \bmod q$$

Each side keeps the X value private and makes the Y value available publicly to the other side. User A computes the key as

$$K = (Y_B)^{X_A} \bmod q$$

and user B computes the key as

$$K = (Y_A)^{X_B} \bmod q$$

5.7.3 Eliptic Curve

Eliptic Curve Cryptography (ECC) is an approach to public key cryptography based on the algebraic structure of eliptic curves over finite fields. The use of eliptic curves in cryptography was suggested independently by Neal Koblitz and Victor S. Miller in 1985[17]. The intricate details and mathematics involved in eliptic curve cryptography are beyond the scope of this thesis. The main reason why we want to mention eliptic curve cryptography is because it has different computational costs compared to other public key methods which can be beneficially in mobile environments[5].

5.7.4 Simple Pairing Objectives - Security

The primary goal of Simple Pairing is to simplify the pairing process from the point of view of the user. Secondary goals are to maintain or improve the security in Bluetooth. Since high levels of security and ease-of-use are often at opposite ends of the spectrum in many technologies and products, much care has been taken to maximize security while minimizing complexity from the end user's point of view[12]. Simple pairing will offer a higher degree of protection against among others:

Passive eavesdropping is one of the more troublesome aspects of wireless communication, and to give a good defence against it we need a strong link key coupled with a strong encryption algorithm. The strength of the link key is based on the amount of entropy or randomness in the generation process of the key, and in legacy pairing [15], the only source of entropy is the Personal Identification Number (PIN). In a normal user environment this is a 4-digit code either supplied by the user, or fixed for a given product. To attack this security measure,

[5]for more information on this topic see: N. Koblitz, Eliptic curve cryptosystems, in Mathematics of Computation 48, 1987, pp. 203-209

one could normally run an exhaustive search for the PIN using commonly available computing hardware [12]. Since simple pairing will implement public key cryptography, this recording attack will become much harder as the attacker must solve a public key problem in order to derive the link key from the recorded information. Simple pairing uses Eliptic Curve Diffie-Hellman (ECDH) key exchange (see section 5.7.3 and section 5.7.2)

Man-in-the-middle attack occurs when a user wants to connect two devices but instead of connecting directly with each other they unknowingly connect to a third (attacking) device that plays the role of the device they are attempting to pair with. The third device then relays information between the two devices giving the illusion that they are directly connected. By doing this, the attacker can eavesdrop on every stage of the communication, obtain disclosed information or passwords, or he can modify the data transmitted to create confusion or change the nature of a transaction beeing made[13].

This is the the Simple pairing association model[6] as it is suggested in the simple pairing whitepaper[12].

5.8 Wireless Security - Bluetooth and WIFI

Since attackers no longer had to be physically connected to the network, the wireless communication itself is considered a weakness in the security of these networks.

5.8.1 WEP, WPA and WPA2

The 802.11 standard prescribes a data link-level security protocol called Wired Equivalent Privacy (WEP), which is designed to make the security of a wireless LAN as good as that of a wired LAN. WEP encryption uses a stream cipher based on the RC4 algorithm [19]. RC4 is an algorithm designed by Ronald Rivest, which was considered safe, allthough several papers released in the late 90's tried to prove otherwise by describing algortihms that could be used to break it [18], they where not very efficient when RC4 was used with a reasonable key length, such as 128 bit. However a more serious problem was described in the publication "Weakness in the Key Scheduling

[6]From the simple pairing white paper - Bluetooth Core Specification Working Group http://www.bluetooth.com

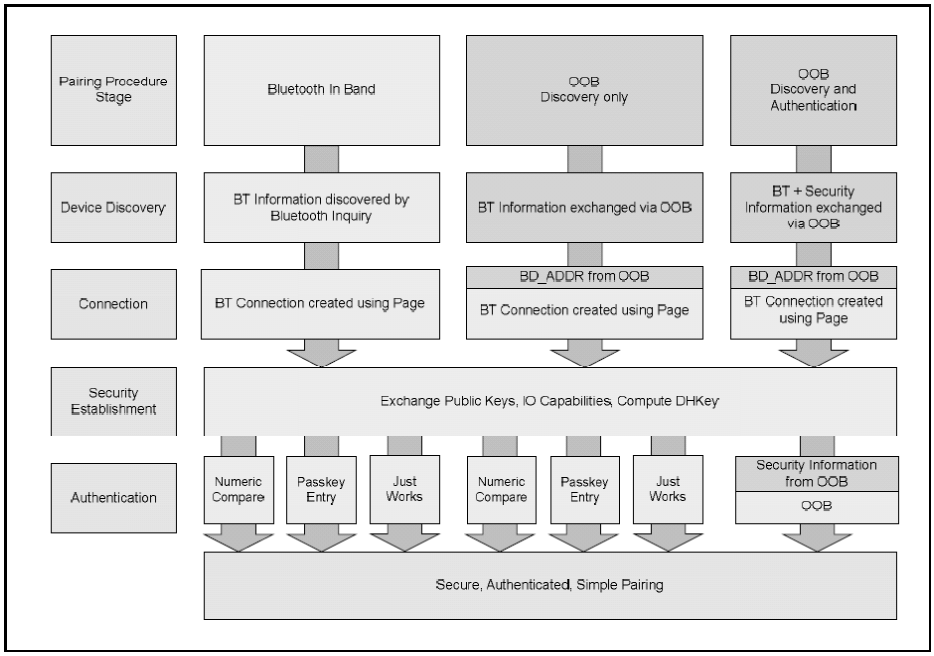

Figure 6: Simple pairing association model[12]

Algorithm of RC4" published in 2001 [8]. The next generation, WIFI Protected Access or WPA was designed as a result to these and other weaknesses in WEP, and it has since been updated again to WPA2 [3][10].

5.8.2 Bluetooth Security

Instead of an extended summary of different sources we see that the presentation provided by the Bluetooth Special Interest Group (SIG) on security is sufficient[7].

Today's wireless world means that data is being sent invisibly from device to device and person to person. This data, in the form of emails, photos, contacts, addresses and more needs to be sent securely.

Bluetooth wireless technology has, from its inception, put an emphasis on security while making connections among devices.

The Bluetooth Special Interest Group (SIG), made up of more than 8,000 members, has a Security Expert Group. It includes engineers from its member companies who provide critical security information and requirements as the Bluetooth wireless specification evolves. Implementing Security Developers that use Bluetooth wireless technology in their products have several

[7]Bluetooth Security Working Group, http://www.bluetooth.com/Bluetooth/Technology/Works/Security/

options for implementing security. And there are three modes of security for Bluetooth access between two devices:

- Security Mode 1: non-secure
- Security Mode 2: service level enforced security
- Security Mode 3: link level enforced security

The manufacturer of each product determines these security modes. Devices and services have different security levels. For devices, there are two levels: "trusted device" and "untrusted device". A trusted device has already been paired with one of your other devices, and has unrestricted access to all services.

Services have three security levels:

- Services that require authorization and authentication
- Services that require authentication only
- Services that are open to all devices

Misinformation Surrounding Security There has been some confusion and misinformation surrounding security and Bluetooth wireless technology.

The reality is the encryption algorithm in the Bluetooth specifications is secure. This includes not just mobile phones that use Bluetooth technology, but also devices such as mice and keyboards connecting to a PC, a mobile phone synchronizing with a PC, and a PDA using a mobile phone as a modem, to name a few of the many use cases.

Cases where data has been compromised on mobile phones are the result of implementation issues. The Bluetooth SIG diligently works with members to investigate any issues that are reported to understand the root cause of the issue.

If it is a specification issue, the people at Bluetooth SIG will work with members to create patches and ensure future devices don't suffer the same vulnerability. This is an on-going process. The recently reported issues of advanced "hackers" gaining access to information stored on select mobile phones using Bluetooth functionality are due to incorrect implementation.

6 Easy Pairing in Practice

The main objective of the easy pairing procedure is to make communication setup easier for the users. In order to do this we have used several different types of technology, including NFC, RFID and Bluetooth. In order to establish a Bluetooth connection (pairing) the Bluetooth stack makes use of the hardware or MAC address of the Bluetooth hardware. For client A to connect to client B, or initiate contact, client A will first start a scanning procedure to discover nearby devices. The main idea behind this easy pairing implementation is to embed the Bluetooth MAC adress and additional security information to a RFID chip, making it possible to read from NFC capable devices. When the RFID chip has been detected, we trigger a Bluetooth scan, and search for the device matching this information. If this device is found, we initiate the connection using the additional security information as parameters. This information can be a pin number, but can also indicate other parameters used in the connection procedure. These include deciding which device should be the master or slave of this session, encryption details and naming information. In a public access point environment we can embed encryption keys or WPA passphrases on the RFID chip, and automatic initiate a WIFI connection when you scan the chip using your PocketPC or laptop.

As we have described in the scenario, easy pairing can be an entry point for personalized service access.

6.1 Easy Connection Set-up

We will now go through the process of establishing the easy connection setup and in that process show some figures of runtime examples and equipment. The setup will be explained in 4 steps, each step showing the underlying technology and user interaction to complete the step.

1. We use a media server to act as the service provider for this setup. Since we want the server to initiate and act as the master of this connection we install the scr-3310 contactless reader and load the media server with the server part of the easy connection software. As shown in figure 7.

   The server software will establish a connection to the reader and start up the Bluetooth sub system. It then listens for incomming connection on the contactless reader.

2. Instead of limiting this setup process to include only the NFC capable phones, we decided to create a solution which could infact work on any

Figure 7: Media server with contactless reader

Bluetooth enabled device. To do so we used RFID chips which were encoded with the information required by the server to successfully establish a connection. The information stored on the chips were a identification string, letting the server know that this in fact was a correctly formated RFID chip indicating a request for easy pairing. This was followed by the hardware address of the Bluetooth interface of the client, and a paramterlist including a PIN code to be used in the pairing procedure. In figure 8 we can see a picture of the two different RFID chips we had available.

For this project the encoding of the RFID chips was done manually, but this procedure can be automated by creating a progam which extracts the nesseccary information from the Bluetooth device and writes it to the RFID chip.

3. To start the pairing procedure, we move the client device into the vicinity of the contactless reader attached to the server. This is showned in figure 10.

   The server will detect that a device is requesting to initiate an easy pairing and start scanning for nearby Bluetooth devices. To do this the server software must implement the Discovery Listener Interface, and the 4 callback methods listed in this interface. When a device is

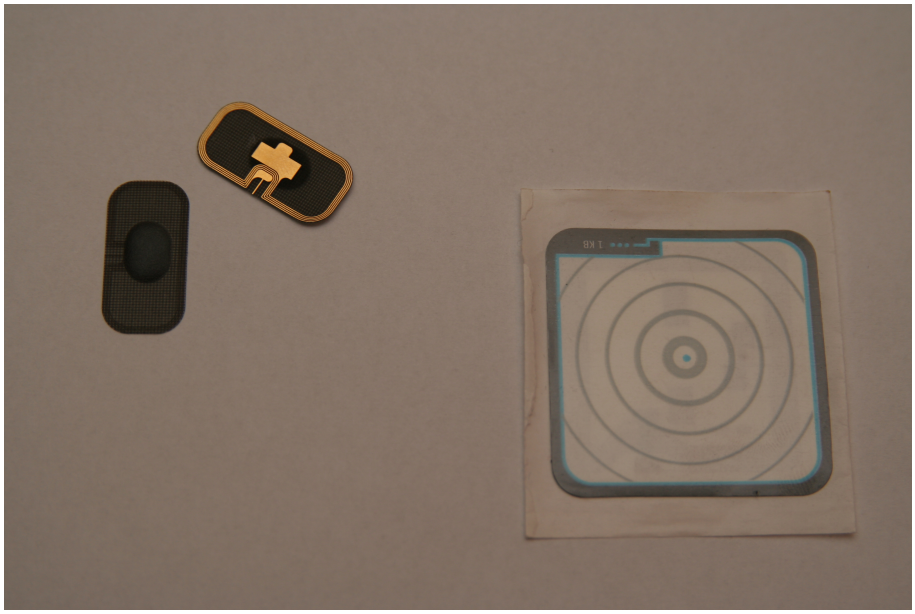

Figure 8: By attaching these chips to a Bluetooth enabled device we can enable easy pairing.

Figure 9: Nokia 6131 and RFID chip

Figure 10: Triggering easy pairing by placing phone next to contactless reader

discovered, it will trigger the deviceDiscovered method. Other methods are listed in the appendix.

This puts every discovered device into a list, which we then can search to see if the device matching the information encoded on the RFID chip is nearby. If the device can not be found, the system will report this and reset itself. If the device is found we continue by initiating a search for services on this device. The service examples are listed in section 6.2 on the next page.

```
public void deviceDiscovered(
            javax.bluetooth.RemoteDevice remoteDevice,
            javax.bluetooth.DeviceClass deviceClass) {
        System.out.println("BTMIDlet.deviceDiscovered");
        // Keep track of discovered remote devices
        discoveredDevices.put(
            remoteDevice.getBluetoothAddress(), remoteDevice);
    }
```

Figure 11: Method to handle discovered devices

Figure 12: Requesting pin input from user to complete pairing

If the server is running Linux, we must interact directly with the Bluetooth subsystem to enable the pairing procedure without demanding any user interaction on the server. To establish this we programmed our application to rewrite the bluez configuration file and update it with the pin code that was included on the RFID chip. When we start the pairing, the server will use this pin as a part of the pairing procedure without prompting for additional input.

4. At the client side of the connection, the previous step will initiate a request for pairing notification. For the user this will seem like a normal pairing procedure as we can see from figure 12, and the user is presented with a pin request. To complete the pairing, the user must issue his or her pin code, the same pin that was written to the RFID chip at an earlier stage.

   The pairing procedure is now complete and we can proceed by choosing one of the service examples explained in the next section.

6.2 Service Examples

With this in place, we can describe some service examples. The following sections will give a short overview of some of the more typical usage scenarios

which we have been testing.

6.2.1 Bookmarks and Shortcut Link Distribution

After the initial contact, the server will act as a service provider and find the best way to distribute content. If this content is distributed using a web-interface, the server will proceed with a push2phone procedure and upload a link to this page directly to the user. The following steps are necessary:

1. Initial contact setup
2. Authentication
3. Create personal link
4. Push link information to mobile equipment

When the user accesses this link, he or she get the option of authenticate using a pre-configured username and password or we can enable a single sign-on feature by including the authentication procedure in the link produced from the server, using the user information exchanged in the pairing procedure as means for further authentication.

6.2.2 Client Software Upload

If the content available in this domain requires additional software, the server will initiate an upload of this software to the mobile equipment. Because there are so many different mobile equipment software solutions, this creates the need to add some additional steps:

1. Initial contact setup
2. Authentication
3. **Request information from mobile equipment**
4. **Choose the appropriate software for this model**
5. Push software to mobile equipment

With this approach we can create complete software solutions, with client software distributed to any platform without demanding too much of the users, making it in essence an easy application distribution solution.

6.2.3 Authentication and Access Control

Another service example is the mobile-key solution. This provides the means for using your mobile device as a key, providing access to otherwise restricted areas. By adding wireless communication software like a contacless reader to the office door, we can authorize a user directly or let the user initiate a key request from the server,

1. Initial contact setup
2. Authentication
3. **Create security certificate for access**
4. Push certificate to mobile equipment

When the user has been authenticated he or she recieves a certificate which he or she can use to gain access to an office building or login to a secure terminal using a mobile device as key. [8]

6.3 Media Center

A media center is a computer adapted for playing music, watching movies and viewing pictures stored on a local hard drive or on a (sometimes wireless) network, watching DVD movies and often for watching and recording television broadcasts. Some software is capable of doing other tasks, such as finding news (RSS-feeds) from the Internet. Media centers are often operated with a remote control, connected to a television set for video output, and can often function as a normal personal computer. A media center can be purpose-built, or created by individuals by adding media center software to a PC or some other computer.

Typical, complete media centers offer the following functions to the user:

- Integration of all forms of media, entertainment and communication functions including TV-reception (analogue TV, DigitalTV via terrestrial-, cable-, satellite-, IPTV-, webTV-networks), broadband
- Internet access, IP-telephony, video-telephony, e-mail etc. into one common user friendly GUI (graphical user interface) controlled with a remote control or wireless keyboard by the family members typically in the living room

[8] Further reading - The mobile phone as Doorkeeper - Thomas Halvorsen - http://wiki.unik.no/index.php/Main/RFID-Doorkeeper

- Ability to receive, store and play back digital media files (via direct video signal, computer network or USB)
- Simplicity
- Cost saving
- Portability compared to a computer equipped to accomplish transfer, storage and TV/hi-fi playback.

If we can add simple pairing to this equation, we can provide content access to mobile users and in order to examine this in more detail we have chosen to describe a media center solution built on the Linux platform.

6.4 A Linux Based Appliance

As a difference compared to the desktop market, in the media player world, the different Unix/Linux variants are dominating, especially on the budget side. In the following, we present a special appliance, a Linux based satellite receiver.

The target customer of the Dreambox 7025[4] is the early adopter with deep technical knowledge. The basic functions are available as usual: the initial setup is done in several minutes and the user is getting a picture. In order to enable the advanced functions, the hardware is up to date:

- a 300MHz ATi Xilleon MIPS SoC,
- 128 MB RAM,
- 32 MB flash,
- 2 independent tuners,
- ethernet, USB and Compact Flash connectors,
- Internal IDE hard drive bay
- Two conditional access card slots.

Only two features are missing compared to the state of the art: a high resolution display connector (HDMI/DVI) and MPEG4 decoding capability, which functions will be added in the next hardware version. To serve the target user group, the manufacturer allows internal hardware modifications

and third party firmwares. To enable easy customisation, the operating system is open source, the graphical interface is implemented in Perl and user data is stored in standard XML files.

The unit supports the usual Conditional Access (CA) solutions and enables the use of multiple subscription cards. By default, two card readers and one Conditional Access Module (CAM) slot is available for smartcards and for future extensions.

The real capabilities of the unit are only revealed if it is connected to a LAN. As this is the case at early adopters, the Dreambox can provide a mixture of server and client services. The only resource, which is limited out-of-box is the storage space, which can be solved easily.

6.4.1 Hardware Extensions

The unit offers three methods for connecting local storage:

USB Standard USB ports, supporting mass storage devices (both flash and hard drives).

CF CF type II slot, supporting high capacity cards.

IDE Internal device bay for 3,5 inch hard drives.

The local disks are mounted in the local Unix file system and are accessible transparently. The three methods are supporting different needs. The card reader slot offers a low power, low noise solution for storing extensions and swap, the internal hard drive bay enables standalone PVR functionality (drives over 200 GB are supported), where only heat generation can be a problem. The CF connector is only used for storage although the connector is electrically identical with the PcCard thus theoretically allowing a wide range of devices to be connected.

The USB connector allows the user to attach both flash disks (same functionality as the CF slot) or external hard disks (same as the internal bay, but without heat problems) with the only drawback of requiring external space.

While PVR functionality is not exceptional on the market, the way this unit provides it differs: it allows the storage of Free To Air (FTA) and encrypted channels and manages the streams on the transponder level, which allows parallel recording or multi-channel Picture in Picture (PiP) on a single tuner.

As local storage is appealing for standalone operation, in most setups storage is provided over the LAN.

6.4.2 Network Connectivity

The device has an integrated 100Base-T ethernet card. Although it provides lower bandwidth, than the internal hard disk, offers a more flexible solution. The advantages of the Linux operating system are clearly visible in this area, as the unit supports the most important network file systems:

CIFS Common Internet File System, formerly Small Message Block (SMB), the standard network file system on Windows based devices

NFS Network File System, standard Unix network file system, Internet Engineering Task Force (IETF) standard.

Other As an addition, FTP, SFTP and SCP protocols are also supported.

The box can act both as client or server, either providing access to the integrated harddisk (where no network storage server is present) or mounting a network share instead of the internal storage (to enable the use of a local disk array or storage unit). Since the Dreambox supports both NFS and CIFS, practically all desktop operating systems can be a counterpart of the Dreambox.

From the network services viewpoint, the unit looks like more a normal, bit slow Linux server, but in the services, which are offered over the local network, the media services are coming visible.

6.4.3 Local Services

Services for local use share two common features: no bandwidth limitation and usually lack of authentication. This enables convenient use and with correctly configured firewall (which is the case because of the customer knowledge), does not compromise security. On this level, the box is starting to show two faces, one following the Linux like behaviour and one playing as a Set-Top-Box (STB).

The services are built on the networking capabilities, and offer

SSH Secure SHell for remote access,

UPnP Universal Plug and Play server and client for service discovery and advertisement,

Cron for task scheduling,

Web interface to enable access to all functions from a remote client (both server and STB capabilites),

Web server to act as a simple web server

Print Server or scan server, where the units can be connected to the USB port.

With these features, the unit can act as a mini server if needed, but more typically, this allows a tight integration with the existing network infrastructure and the use of standard and secure methods for remote access.

Since this device is primarily a fully featured Set-Top-Box (STB), beside of viewing content on the TV screen, it offers

Timeshift to pause live tv streams,

Recording to record programs from any channel,

Skins to enable user interface customisation

Flexible tuners and interchangeable tuners for both satellite, cable and terrestrial reception.

The interface is programmed in Perl and can be modified by the user. All Set-Top-Box functions can be accessed via the web interface, and are sending back an XML file as a result. This functionality enables automatic error handling, as error messages are sent in a well defined, machine readable form.

6.4.4 Streaming

As seen in the sections before, the Dreambox usually may act both as server and client. Also in streaming services, where it offers an integrated client for web radios, MPEG2 streaming content from a network server and direct player for MPEG2 TS files.

Remote access becomes more important, if the user wants to use the serving capabilities of the unit. By default, the Dreambox is capable of streaming out the original MPEG2 stream over the network interface and can act as a local IPTV head end. With this functionality, the users can deploy an own IPTV solution on their LAN and also small cable-tv networks can be exchanged with an ethernet based solution (because of the fact, that the unit is capable of streaming out all channels coming on the same transponder).

If a network server with appropriate resources is available, the transcoded video can be streamed over the internet to the user, if he is in a remote location. After the IP based access is enabled to the content, the services provided by the Dreambox can be integrated into other media center solutions, which offer complementary features.

6.4.5 Easy Pairing on Dreambox

Since the Dreambox basically is a normal computer running Linux, we used the built in Linux Bluetooth Stack in cooperation with our software to implement simple pairing. We then used the Link distribution example and the integrated web interface on the dreambox to allow remote control. Some of the experiences we gained in this process is discussed in Implementation Issues in section 7.

6.4.6 Drawbacks

Our study shows, that while the Dreambox is capable of serving as a decent STB and offer easy pairing capabilities, the Perl interface is unstable, the unit is unable to render MPEG4 content and the lack of HDMI/DVI connectors limit its HD capabilities.

This problems can be solved effectively, if the user deploys the Dreambox as a server. In this case, the unit will act as an IPTV server. With open source plugins, the remote control of the unit can be integrated into media center solution.

During our study, we faced several interesting problems, where the local hard drive's problem was the most interesting. As one of the first steps, a small, 20 GB unit was installed into the dreambox to provide appropriate local storage for recording and time shifting tests. The benchmark of the unit showed a good transfer rate of over 16 MB/s, where the expected maximum data rate from the satellite feeds weren't exceeding the 5 MB/sec range. The hard disk was rotating with 5400 RPMm, to produce less heat than today's 7200 RPM disks, as this was a known problem with earlier versions of the Dreambox, which suffered from overheating problems in the presence of a high performance winchester. However, the record tests already showed several skips in the streams and the time shift feature was practically unusable.

We made an extensive investigation, which revealed, that the skips also coincide with loads of over 1,3 on the box. After various checks, we concluded, that the problem lies in the IDE module and may be caused by some compatibility problems between the IDE controller and the old 5400 RPM disk, which was forced to PIO mode instead of DMA. This resulted in a maximum write performance of less than 1 MB/sec. This problem was surprising, as the unit was able to test the drive and give a result of over 16 MB/sec in benchmark mode.

In the period, when the box was used with the internal disk, the graphical interface was unstable and needed a restart every day. This problem was solved with the change to network storage, although only the graphical

interface suffered and the rest of the operating system was working without any sign of heat problems.

A more generic problem is the slowness of the Perl based GUI and the long boot time of over a minute. This is particularly a problem since the standby mode is more of a fake, since the box is only muting the video and audio outputs, but actually proceeds to run and decodes the stream continuously. Our power consumption measurements prove this problem, as we noticed, that the consumption changed only insignificantly if going to stand-by.

The problems with the user interface lead us to a conclusion, that this unit is at the moment more suitable as a background server, because of the stability of the operating system. This task can be easily done with the existing plugins both for Linux and Windows based media centers.

6.5 Bluetooth PAN and Personal Content Access

One of the more interesting examples comes from introducing PAN as the next step after simple pairing. Most of the Bluetooth enabled devices used in this work have support for PAN and were able to run a TCP/IP connection over the Bluetooth link. If we add the following steps to our procedure:

1. Initial contact setup
2. Authentication
3. **Bluetooth Service Discovery search**
4. **If present, request connection to PAN-access service**
5. **Connection established, and all further communication go over TCP/IP**

We can use this connection as it is or we can combine this with the bookmarks and link distribution or client software upload. Since this is basically a local ip network connection between the devices, it is ideal for home system environments. Since this procedure establishes a TCP/IP connection between the devices, we are also given the possibility of connecting to other devices on the same subnet. Instead of connecting to one device at a time, to use the services provided by that particular device, we can introduce one new device into an already working environment and make its services available for all the connected devices without having to go through an advanced setup routine. To elaborate a little bit on this subject, one could setup a complete environment using a mobile key solution in order to gain access to a room

and in that process authenticate and enable access to other services located within this room, configurated to suit the users needs. In practice this would be a single sign on procedure to your own home and appliances.

6.6 Personal Content Access through PAN in relation to Windows Media Center

As we have mentioned earlier in 2, Microsoft introduced a plugin for the Windows Media center edition, which can allow Personalised Content Access. This plugin is called WebGuide[9] and has been developed independently by Doug Berrett until Microsoft bought it and made it publicly available September 2007. WebGuide allows you to remotely view live and recorded TV programs and to remotely schedule and manage your recorded television programs, music, pictures and videos on your Media Center or Windows Vista PC. It operates directly on the machine working as a media content provider and is compatible with Windows media center edition, Windows Vista and the newly released Windows Home Server. To access it we can use a normal web browser. Users are requested to authenticate in order to gain access, and then it provides the user with a web-interface which is very similar to the one of a normal Windows media center. As shown in figure 13.

The user authentication can be automated when using the Link Distribution described in section 6.2.1.

Webguide also has a mobile user interface which is designed to work on cellular phones and PDAs. By using this interface we can schedule and change recordings and on some models it is also possible to gain access to the content available on the media center unit and stream it directly to the mobile equipment.

[9]Webguide can be found at http://www.asciiexpress.com/webguide/

Figure 13: Webguide user interface screenshot taken from the WebGuide Documentation

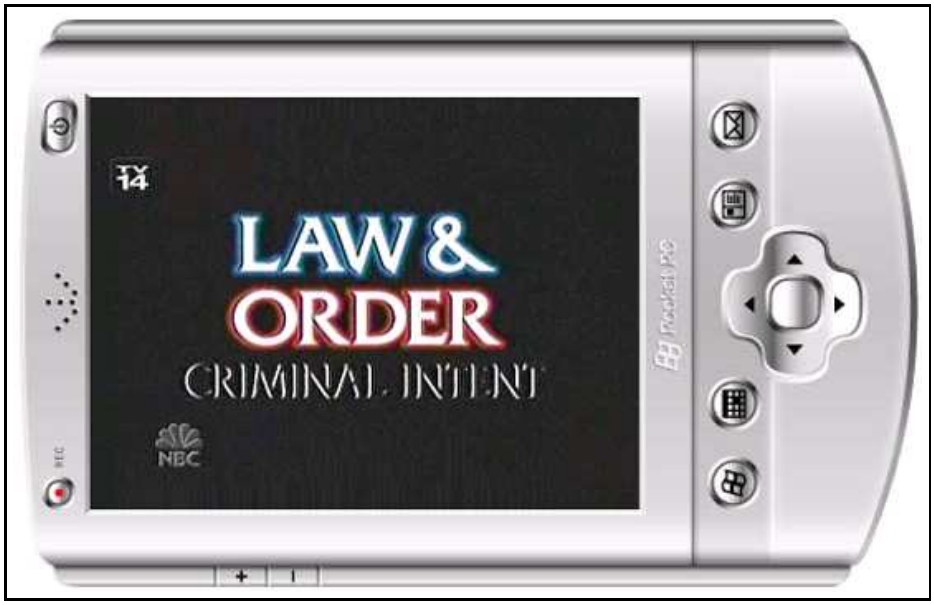

Figure 14: Streaming video from media center to a PPC

7 Implementation Issues

As we have seen from the previous sections, the easy pairing is supposed to be as simple as possible for the users, but while researching and developing this solution we found some difficulties which will be adressed in this section.

7.1 Pin Request Dialog using Java, Bluetooth and Windows

As a security measurement, the java Bluetooth stack in Windows will demand user input to provide a shared passphrase or PIN code in order to complete a pairing procedure. A lot of our research was put into automating this process directly from java code, but for Windows this was unsuccessful. In Linux we were able to bypass this by accessing the Bluetooth subsystem configuration files directly and updating the security information directly. This is somewhat of a work around and a solution to this problem should be investigated further.

7.2 Security Risks in Current Implementation

Since this solution was implemented using clear text information stored on the RFID chips, and we also included the PIN used to secure the connection into this data, this is not a very secure system. The only thing preventing attackers from obtaining this data, is the limited range of the contactless technology. Therefore we see the need to add another security challenge in form of a login page before giving access to personalized content.

7.3 NFC Tags and Devices

During the switch from the Nokia 3320, to the new and improved Nokia 6131NFC, we discovered that the new phone lacked support for the GemPlus phase 1 RFID cards. This is probably due to the fact that Nokia upgraded the firmware on the newer model, and have left out support for earlier RFID tags. The tags we used are showed in figure 8 on page 28. The MiFare classic on the left and the (unsupported) GemPlus on the right.

8 Conclusion

This work adressed interworking of devices in the home network. It is based on the assumption of a mobile device being the key element of the security infrastructure. The mobile phone is used to distributed access keys to other home devices. Having enabled communication, the mobile phone might act as remote control or client for service access.

The thesis includes the following elements. It reviews the state of the art of technology, then defines user scenarios for the home user and explore a solution for easy pairing. Our recommendation is to use Near Field Communication (NFC) to initate the communication between devices. We see Bluetooth and Wifi as technologies for device interworking at home. We identified interworking to selfstanding devices such as the car, integration of multimedia devices such as set-top-box and secure authentication for payment and access control. We established Bluetooth pairing for a content access scenario, a Linux based set-top-box. We also outlined the method required for WIFI pairing.

Our experience shows that the technology is mature and that such a service can be introduced with a low cost, leading to a better user experience and may enable transparent deployment of complex services without requiring advanced technological knowledge from the end-user. During our research process we have seen that there could be several different approaches to this problem, and we clearly see the need for industrial standardisation to ensure interoperability between the different device categories.

9 Reference

References

[1] 802.11b whitepaper
http://www.vocal.com/white_paper/ieee_802.11b_wp1pdf.pdf.

[2] Alvarion wimax whitepaper
http://www.wimax-industry.com/wp/papers/alvarion_wimax_wp.pdf.

[3] Ieee 802.11i specification
http://standards.ieee.org/getieee802/download/802.11i-2004.pdf.

[4] Dreambox 7025 user manual, 2006.

[5] T. Baidas, M. W.; Oapos;Farrell. The performance of coded non-coherent m-ary orthogonal keying based ofdm systems in a frequency selective and fast time-varying channel.

[6] Paul J.Kline C Bala Kumar and Timothy J. Thompson. *Bluetooth application programming with the java API.* Morgan Kauffmann publishers, 2004.

[7] W Diffie and M. Hellman. Multiuser cryptographic techniques. 1976.

[8] S. Mantin I. Fluhrer and SHamir A. *Weakness in the key scheulding algorithm of RC4.* Proceedings, workshop in Selected Areas of cryptography, 2001.

[9] Louis E. Frenzel. Nfc makes great progress in the wireless world. *Electronic Design*, 2006.

[10] A Gilbert. *Wireless security study guide.* Distributed Systems Online, IEEE Volume 6, Issue 11, 2005.

[11] Bluetooth Core Specification Working Group. Bluetooth core specification 2.1 + edr
http://bluetooth.com/bluetooth/technology/works/core_specification_v21__edr.htm, 2007.

[12] Core Specifiaction Working Group. Simple pairing whitepaper
http://bluetooth.com/nr/rdonlyres/0a0b3f36-d15f-4470-85a6-f2ccfa26f70f/0/simplepairing_wp_v10r00.pdf. 2006.

[13] Hypponen Konstantin Haataja and Keijo M.J. *"Nino" man-in-the-middle attack on bluetooth secure simple pairing.* Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia, 2007.

[14] Thomas Halvorsen and Haakon Eikenes. Mobile key. Master's thesis, UiO, kjeller, 2006.

[15] Bluetooth Special interest group. Bluetooth core specification 2.0.

[16] C.S.R Prabhu and A. Prathap Reddi. *Bluetooth technology.* Prentice-Hall Of India Pvt. Ltd., 2004.

[17] K.Hideki Imai SeongHan Shin, Kobara. *Elliptic Curve based Authenticated Key Agreement Protocol for Wireless Security.* Computational Intelligence and Security International conference, 2006.

[18] William Stallings. *Cryptography and network security - principles and practices Third edition.*

[19] Andrew S. Tanenbaum. *Computer networks 4th. edition.*

[20] Telenor Tvillingkort. http://www.telenor.no/privat/mobil/tjenester/mobilnummer/tvilling/index.jsp.

[21] W.Rankl and W.Effing. *Smart Card Handbook.* John Wiley and Sons, 1997.

[22] Nfc forum - http://www.nfc-forum.org/home.

[23] Specification of the bluetooth system - core, version 1.1.